Automatic Calibration of Multiple Coplanar Sensors

Jonathan Brookshire and Seth Teller

Computer Science and Artificial Intelligence Laboratory

Massachusetts Institute of Technology

Cambridge, MA 02139

Email: {jbrooksh,teller}@csail.mit.edu

Abstract—**This paper describes an algorithm for recovering the rigid 3-DOF transformation (offset and rotation) between pairs of sensors mounted rigidly in a common plane on a mobile robot. The algorithm requires only a set of sensor observations made as the robot moves along a suitable path. Our method does not require synchronized sensors; nor does it require complete metrical reconstruction of the environment or the sensor path. We show that incremental pose measurements alone are sufficient to recover sensor calibration through nonlinear least squares estimation. We use the Fisher Information Matrix to compute a Cramer-Rao lower bound (CRLB) for the resulting calibration.**

Applying the algorithm in practice requires a non-degenerate motion path, a principled procedure for estimating per-sensor pose displacements and their covariances, a way to temporally resample asynchronous sensor data, and a way to assess the quality of the recovered calibration. We give constructive methods for each step. We demonstrate and validate the end-to-end calibration procedure for both simulated and real LIDAR and inertial data, achieving CRLBs, and corresponding calibrations, accurate to millimeters and milliradians. Source code is available from http://rvsn.csail.mit.edu/calibration.

I. INTRODUCTION

Robotic platforms often employ multiple proprioceptive and exteroceptive sensors. For many applications, the data from such sensors must be fused spatiotemporally, requiring that the relative pose of the sensors be estimated. For example, running a grid-based obstacle avoidance algorithm on a robot with two planar LIDARs would require the relative pose of the LIDARs to fuse their scans into one grid. A Simultaneous Localization and Mapping (SLAM) algorithm might use the relative pose of a LIDAR and an inertial measurement unit (IMU) to express their motions in a common reference frame.

This paper describes a method to estimate the 2D calibration between any pair of coplanar sensors using information easily gathered in the field. The input consists of per-sensor incremental pose estimates (and corresponding error bounds) as the robot moves along a suitable path. The output is a 3-DOF (i.e., (x, y, θ)) calibration for each sensor pair, along with a lower bound on the error of the recovered calibration.

A natural question is whether incremental pose observations alone contain sufficient information to recover relative sensor calibration. Intuition suggests that such observations are indeed sufficient (Fig. 1). The pose difference between two sensors causes them to experience different incremental motion (Fig. 1b and 1c). The calibration that best transforms the motion of the first sensor to align with that of the second sensor will be the desired rigid-body transformation.
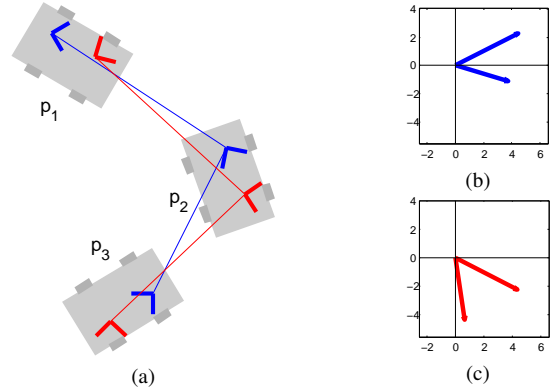

Fig. 1: As the robot moves from p_1 to p_3, two sensors, positioned as in (a), will experience different translational ((b) and (c)) and rotational (not shown) incremental motion. The calibration relating the sensors is the transformation that best brings the disparate observed motions into agreement.

Below, we formalize this intuitive argument and develop a procedure to estimate the calibration. We show that the calibration is observable (i.e., that there exists sufficient information in the input observations to estimate the calibration), so long as certain degenerate motion paths are avoided. We show how to compute the Cramer Rao Lower Bound (CRLB) for the calibration estimate. The CRLB provides a best-case minimum error (covariance matrix) for the estimate, enabling the accuracy of the calibration to be assessed. The CRLB covariance can also be integrated, as a source of uncertainty, into subsequent processing.

Our method is applicable to any sensor, or sensor combination, that permits observation of its own relative motion. The process can also be applied to find the pose of sensors relative to a robot frame, if the relative motion of the robot body can be observed (e.g., using inertial and odometry data). The calibration procedure can be summarized as:

1) Drive the robot along a non-degenerate planar path.
2) Recover per-sensor incremental poses and covariances.
3) Resample asynchronous sensor data.
4) Use least squares to recover relative sensor calibration.
5) Compute the CRLB to compare the actual estimate to the best-case achievable accuracy.

After formally stating the calibration problem (§ III), we show that calibration is observable (§ IV). We describe the estimation process and bias computation in § V and § VI,

respectively. In practice, one typically must (1) interpolate incremental poses (§ VII) and their associated covariances for asynchronous sensor data and (2) reliably estimate the covariances for relative pose observations for common sensor such as LIDARs (§ VIII). Using simulated and real data from commodity LIDAR and inertial sensors, we demonstrate calibrations accurate to millimeters and milliradians (§ IX).

II. Background

Ideally, it would be easy to determine the transformation relating any pair of sensors. One could use a ruler to measure the translational offset, and a protractor to measure the orientation offset. In practice, it is rarely so easy. A sensor's coordinate origin typically lies inside the body of the sensor, inaccessible to direct measurement. Curved and textured sensor housings, and obstructions due to the robot itself, usually make direct measurements of sensor offsets difficult or impossible.

Alternatively, one could engineer the calibration problem away by designing and fabricating precision sensor mounts. While this approach may be practical for small, closely-spaced sensors such as stereo rigs, it would be cumbersome and costly for larger systems. Moreover, the need for machined mounts would hinder field adjustment of sensor configurations.

In light of these practical considerations, we desire a software-only algorithm to estimate calibration using only data easy to collect in the field. One approach would be to establish a common global reference frame, then estimate the absolute pose of each sensor at every sample time relative to that frame; the calibration would then be the transform that best aligns these poses. For example, in the setting of Fig. 1, the task would be to find the calibration that aligns the blue and red sensor paths. In practice, however, recovering accurate absolute sensor pose is difficult, either because of the effort required to establish a common frame (e.g., through SLAM), or the error that accumulates (e.g., from odometry or inertial drift) when expressing all robot motion in such a frame.

Researchers have, nevertheless, attempted to recover absolute pose through various localization methods. Jones [9] and Kelly [11] added calibration parameters to the state of an Extended Kalman Filter and an Unscented Kalman Filter (UKF), respectively. Gao [8] proposed a Kalman filter incorporating GPS, IMU, and odometry to estimate pose. Levinson [12] recovered calibration through alignment of sensed 3D surfaces. Any calibration method employing global, metrical reconstruction will necessarily invoke SLAM as a subroutine, thus introducing all the well-known challenges of solving SLAM robustly (including data association, covariance estimation, filter consistency, and loop closure).

When it is possible to augment or prepare the environment, absolute pose can also be established with known, calibrated landmarks. For example, Blanco [3] and Ceriani [6] used GPS and calibrated rigs to establish a global sensor reference frame.

Our algorithm avoids absolute poses and environment preparation. It requires only per-sensor incremental poses, estimated using a sensor-specific method: for LIDARs or stereo cameras, through scan matching [14] or iterative closest point [2]

methods; for IMUs, through integration of acceleration; for wheel encoders, through accumulation. Although our method may require data association, it does so over shorter time periods than full-blown SLAM. As a result, it can be expected to be more robust and generally applicable than a method depending on SLAM. Censi [5] also used differential poses to recover calibration parameters; our method is distinct in that it accounts for observation noise in a principled way.

III. Problem Statement

Our task is to estimate the static calibration parameter vector $k = [x_s, y_s, \theta_s]$ representing the translational and rotational offsets from one sensor, r, to a second sensor, s. Fig. 2 shows the graphical model relating the variables of interest, with v_{ri} the latent, true incremental poses of sensor r at time i and z_{ri} and z_{si} the corresponding observed incremental poses of the sensors. Since z_{si} depends only on v_{ri} and the rigid body transform k, the true motion of sensor s need not be estimated. Further, let $z_r = [z_{r1}, z_{r2}, \cdots, z_{rN}]$, $z_s = [z_{s1}, z_{s2}, \cdots, z_{sN}]$, and $z = [z_r, z_s]$.
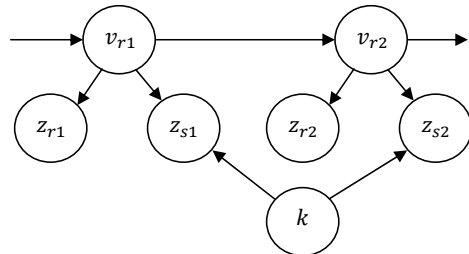

Fig. 2: Graphical model of calibration and incremental poses

Although we are interested only in the calibration parameters k, the true incremental poses $v_r = [v_{r1}, v_{r2}, \cdots, v_{rN}]$ at every sample time are also unknown variables. So the parameters to estimate are $x = [v_r, k] = [v_{r1}, v_{r2}, \cdots, v_{rN}, x_s, y_s, \theta_s]$. Applied naïvely, this state vector has the potential to grow rapidly, adding three terms (two translations and one rotation) for each observation. (However, as we demonstrate with real data in § IX-B, the estimates converge quickly with manageable N; our experiments used $N = 88$ and $N = 400$.) Finally, we define the equations relating the expected observations and true values as:

$$G(v_r, k) = \begin{bmatrix} v_{r1} & \cdots & v_{rN} & v_{s1} & \cdots & v_{sN} \end{bmatrix}^T \quad (1)$$

Each term z_{ri} is a direct observation of the latent variable v_{ri}, with v_{si} a 3×1 incremental pose for sensor s. If all values are expressed as homogeneous matrices (i.e., K is the homogeneous matrix corresponding to the calibration k), then $V_{si} = K^{-1} V_{ri} K$. Note that G has dimension $6N \times 1$.

IV. Observability & the Cramer-Rao Lower Bound

Before attempting to recover the calibration from the observed incremental pose data, we must first show that there exists sufficient information in the observed data to determine the calibration parameters.

Assume we have some estimate $\hat{x}$ of the true parameters x_0. We wish to know how the estimate varies, so we calculate the covariance $\mathrm{E}\left[(\hat{x}-x_0)(\hat{x}-x_0)^T\right]$. Cramer and Rao [15] showed that this quantity can be no smaller than the inverse of J, the Fisher Information Matrix (FIM):

$$\mathrm{E}\left[(\hat{x}-x_0)(\hat{x}-x_0)^T\right] \geq J^{-1} \qquad (2)$$

Here, as in [1], we abuse notation by writing $A \geq B$ to mean that matrix $A - B$ is positive semi-definite.

If the FIM is non-invertible, the variance of one or more parameters will be unbounded. To illustrate, suppose J were a scalar; then with $J = 2$, the variance would exceed $\frac{1}{2}$. If J were 0 (i.e., rank-deficient), the variance would be infinite, meaning that nothing would be known about the computed parameter. Intuitively, if J is rank-deficient, we will not be able to estimate x. We next show that, in general, J has full rank when two incremental pose observations are available. We also characterize the situations in which J is rank-deficient.

A. The FIM and the Jacobian

The FIM captures information from the observations about the parameters [1]. For an unbiased estimator, the FIM is:

$$J = \mathrm{E}\left[(\nabla_x \ln P(z|x))(\nabla_x \ln P(z|x))^T\right] \qquad (3)$$

In § VI, we estimate the bias and, finding it to be small, use this form of the CRLB as an approximation to the bias-accommodating version found in [15]. Using this version, Wang [16] showed that if $P(z|x) \sim N(x, \Sigma_z)$, then

$$J = J_G^T \Sigma_Z^{-1} J_G \qquad (4)$$

where J_G is the Jacobian of the observation equations G. That is, if the observation noise has a Gaussian distribution, then the FIM can be calculated as a function of the observation Jacobian and observation covariance. This has two implications:

1) Given the covariance and having computed the Jacobian, we have the quantities needed to calculate a lower bound on the noise of our estimate.
2) J has full rank if and only if J_G has full column rank [16]. Thus by analyzing the column rank of J_G, we can determine whether the calibration is observable.

B. The FIM Rank

First, consider the case where there is one incremental pose observation for each sensor, i.e., $z = [z_{r1}, z_{s1}] = [z_{r1x}, z_{r1y}, z_{r1\theta}, z_{s1x}, z_{s1y}, z_{s1\theta}]$. The parameter vector is then $x = [v_{r1}, k] = [v_{r1x}, v_{r1y}, v_{r1\theta}, x_s, y_s, \theta_s]$. The resulting 6×6 Jacobian matrix has the form below; it is clearly rank-deficient because rows 3 and 6 are the same (the elided values denoted by $\cdots$ are not relevant).

$$J_{G1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad (5)$$

Second, consider the case where there are two incremental pose observations. The observation vector is $z =$ $[z_{r1}, z_{r2}, z_{s1}, z_{s2}]$. The parameter vector is then $x = [v_{r1}, v_{r2}, k]$. The Jacobian is now 12×9 and has the form:

$$J_{G2} = \begin{bmatrix} I_{6 \times 6} & 0_{6 \times 3} \\ \cdots & M_{6 \times 3} \end{bmatrix} \qquad (6)$$

Due to the 6×6 identity matrix, the first 6 columns are linearly independent. In order to show that J_{G2} has full rank (i.e., rank 9), we need only understand – and avoid – the conditions under which the columns of M become linearly dependent.

C. Conditions for FIM Rank-Deficiency

When a and b can be found such that $aM_1 + bM_2 + M_3 = 0$, where M_i is the i-th column of M, J_{G2} will be rank deficient. With $c_s = cos(\theta_s)$, $c_{1s} = cos(v_{r1\theta} - \theta_s)$, $x_1 = v_{r1x}$, etc.,

$$M = \qquad (7)$$

$$\begin{bmatrix} c_{1s}-c_s & -s_{1s}-s_s & y_sc_{1s}+(y_1-y_s)c_s+x_ss_{1s}-(x_1-x_s)s_s \\ s_{1s}+s_s & c_{1s}-c_s & -x_sc_{1s}-(x_1-x_s)c_s+y_ss_{1s}-(y_1-y_s)s_s \\ 0 & 0 & 0 \\ c_{2s}-c_s & -s_{2s}-s_s & y_sc_{2s}+(y_2-y_s)c_s+x_ss_{2s}-(x_2-x_s)s_s \\ s_{2s}+s_s & c_{2s}-c_s & -x_sc_{2s}-(x_2-x_s)c_s+y_ss_{2s}-(y_2-y_s)s_s \\ 0 & 0 & 0 \end{bmatrix}$$

Since the columns of M cannot in general be eliminated, the system is observable. However, it is useful to further consider which specific kinds of observations will cause M to reduce; by knowing the characteristics of paths that do not admit calibration, we can avoid traversing them in practice. There are several types of path degeneracies:

1) Paths for which each sensor observes constant incremental poses. In this case, $\text{rank}(J_{GN}) = \text{rank}(J_{G1})$.
2) Paths for which the robot remains at a fixed orientation with respect to some arbitrary world frame.
3) Paths for which the sensors move on concentric circles.
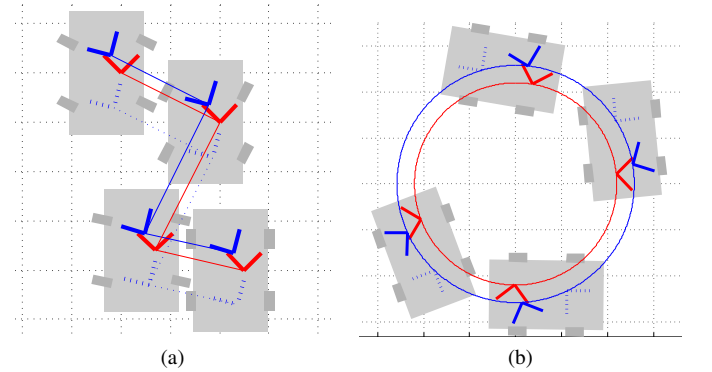

(a)   (b)

Fig. 3: Sensor movements without rotation, as for a translating holonomic robot (a), can prevent calibration. The calibration of a non-holonomic robot, e.g., an automobile, cannot be recovered if the robot is driven in a straight line. Concentric, circular sensor motion (b) can also prevent calibration.

The first case is relatively uninteresting and can be easily avoided in practice by varying the robot's velocity. The second case arises if the robot travels such that the sensors experience only translation (Fig. 3a). In the third case, infinitely many recovered poses for sensor s (e.g., lying along the blue circle

in Fig. 3b) will be consistent with the observed incremental poses. (To see this, compare the relative offsets between pairs of blue sensor frames and pairs of dotted blue sensor frames.) In practice, speed variation alone will not prevent degeneracy; it is the path geometry itself that must be avoided.

D. Other Methods of Showing Observability

Martinelli [13] and Kelly [11] added the static calibration parameters to the dynamic state vector of a Kalman filter. In order to address the question of observability, they used non-linear observability analysis (instead of the FIM) based on Lie derivatives. In their formulations, the robot pose (relative to some arbitrary world frame) is part of the state vector. However, because most sensors make only relative observations, the state is not observable, i.e., there exists an unobservable transform between the robot poses and the world frame. Martinelli overcame this limitation by removing the unobservable portions of the robot pose from the state vector; Kelly overcame it by assuming the sensors can provide world-relative observations. In our formulation, the parameter vector includes only relative motions, so we need not make any simplifications or assumptions to show observability.

V. ESTIMATION

The maximum a posteriori (MAP) estimation is given by:

$$\hat{x}^{MAP}(z) = \underset{x}{\operatorname{argmax}} P(x|z) = \underset{x}{\operatorname{argmax}} P(z|x)P(x) \quad (8)$$

That is, we wish to find the calibration (and true incremental poses) that maximize the probability of the parameters, given the observed incremental poses. Applying the independence assumptions implied by the graphical model in Fig. 2 yields:

$$\hat{x}^{MAP}(z) = \quad (9)$$

$$\underset{x=[v_r,k]}{\operatorname{argmax}} P(v_{r1})P(k) \prod_{i=1}^{N} P(z_{ri}|v_{ri})P(z_{si}|v_{ri},k)P(v_{ri}|v_{r,i-1})$$

In many cases, it may be reasonable to assume that nothing is known about the distribution of the first incremental pose of the sensor, $P(v_{r1})$. If acceleration data or a motion model is available, then the way in which the incremental poses relate, expressed in $P(v_{ri}|v_{r,i-1})$, can be estimated. Similarly, if there is a good initial guess about the calibration itself, then it might be reasonable to assume a Gaussian distribution centered about the guess, reflected in $P(k)$. If a reasonable value for one or more of these distributions cannot be justified, however, then a uniform distribution may be most appropriate. If $P(v_{r1})$, $P(v_{ri}|v_{r,i-1})$ and $P(k)$ are distributed uniformly, they can be removed from the maximization and Equation 9 becomes the maximum likelihood estimator:

$$\hat{x}^{ML}(z) = \underset{x=[v_r,k]}{\operatorname{argmax}} \prod_{i=1}^{N} P(z_{ri}|v_{ri})P(z_{si}|v_{ri},k) \quad (10)$$

Our experiments (§ IX) assume that $P(v_{r1})$, $P(k)$, and $P(v_{ri}|v_{r,i-1})$ are uniformly distributed. In other settings, a normal distribution can straightforwardly be incorporated into the optimization framework shown here.

When the incremental pose observations for sensor r and s, reflected in $P(z_{ri}|v_{ri})$ and $P(z_{si}|v_{ri},k)$, are normally distributed, the optimization becomes a non-linear Least Squares Estimation (LSE). To see this, let $\Sigma = \text{blkdiag}([\Sigma_{r1}, \cdots, \Sigma_{rN}, \Sigma_{s1}, \cdots, \Sigma_{sN}])$ where $P(z_{ri}|v_{ri}) = N(z_{ri}; v_{ri}, \Sigma_{ri})$ and $P(z_{si}|v_{ri},k) = N(z_{si}; g(v_{ri}, k), \Sigma_{si})$, and g is the function that applies the calibration to the sensor r incremental poses to produce the sensor s incremental poses. The blkdiag function arranges the 3×3 covariance matrices along the diagonal of the resulting $6N \times 6N$ covariance Σ. The LSE can then be found by minimizing:

$$\Delta = z - G(v_r, k)$$
$$c(x, z) = \Delta^T \Sigma^{-1} \Delta \quad (11)$$
$$\hat{x}^{LSE}(z) = \underset{x=[v_r,k]}{\operatorname{argmin}} (c(x, z))$$

The LSE cost function $c(x, z)$ returns a single number that can be minimized by many solvers (e.g., Matlab's fminunc function). In practice, we found that solvers that minimize the individual errors directly (e.g., Matlab's lsqnonlin), rather than their summed squares, converge faster. If $\Sigma^{-1} = \Psi\Psi^T$, where Ψ is found e.g. via the lower Cholesky factorization, then

$$c(x, z) = \Delta^T \Psi\Psi^T \Delta = (\Psi^T \Delta)^T (\Psi^T \Delta) \quad (12)$$

The individual errors are now represented in the $6N \times 1$ vector $\Psi^T \Delta$ and solvers such as lsqnonlin can be used.

VI. EVALUATING BIAS

The nonlinear LSE is not necessarily unbiased. That is, the estimate $\hat{x}$ may exhibit an offset from the true value x_0. Using a technique developed by Box [4], the bias can be approximated and, if significant, subtracted from the estimate. We (approximately) evaluate the bias, rather than assume that it is small. In our simulations and experiments, we indeed found it to be negligible compared to the CRLB.

In order to calculate an expected value for the bias, Box approximates it with a second-order Taylor series:

$$\text{E}[\hat{x} - x_0] = \left(\frac{-1}{2}\right) J^{-1} J_G^T \Sigma^{-1} m$$
$$m = \begin{bmatrix} \text{trace}(H_1 J^{-1}) & \cdots & \text{trace}(H_{6N} J^{-1}) \end{bmatrix}^T \quad (13)$$

where J is the FIM, J_G is the Jacobian, and H_i is the Hessian (second derivative) of the i-th observation (i.e., the Hessian of the i-th row in the G matrix). Note that both the bias and the CRLB are driven smaller as $\det(J)$, where J again represents the amount of information in the observations, increases.

VII. INTERPOLATION IN PRACTICE

One requirement of our algorithm is that the relative motions z_r and z_s be observed at identical times. In practice, however, sensors are rarely synchronized. We must therefore interpolate observations and their associated covariance matrices. As shown in Fig. 4, a sensor has traveled along the blue path with several relative motions (blue arrows). In order to synchronize with a second sensor, for example, we wish to calculate the relative motions (black arrows) and covariances at new

times. Although other researchers have developed techniques for interpolating between two covariances [7], we could find no previous work related to interpolating both incremental poses and their covariances. Our key contribution is that the interpolation can be formulated as a function; then, using a sigma point method similar to that used in the UKF, we can estimate resulting motions and covariances.
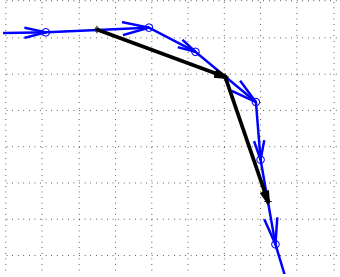

Fig. 4: Resampling incremental poses (blue) at new time steps may make interpolated observations (black) dependent.

Our goal is to take the relative motions, $\overline{v_A}$, and the associated covariance Σ_A, sampled at corresponding times in set A, and interpolate (or resample) them to produce the mean relative motion $\overline{v_B}$ and covariance Σ_B at new times in set B. Note that Σ_A will often be band-diagonal; in other words, the incremental poses observed will be assumed to be independent. On the other hand, Σ_B will most likely not be band-diagonal, since new, adjacent interpolations in $\overline{v_B}$ will likely depend upon the same observation in $\overline{v_A}$ (c.f. Fig. 4), and thus be correlated. Such new dependencies will occur unless $B \subseteq A$.

When the observations of both sensors are resampled at common times, Σ in Equation 11 will take the form $\Sigma = \text{blkdiag}\left(\left[\Sigma'_r, \Sigma'_s\right]\right)$, where Σ'_r and Σ'_s are the resampled covariances of r and s, respectively.

A. The 2D interpolation function

We construct a nonlinear interpolation function, f, of the form: $v_B = f(A, v_A, B)$, where v_A and v_B are $3N \times 1$ vectors of incremental pose observations. Notice that this function does not, itself, estimate the new covariances; instead, it only interpolates the motion. There are many possible ways to design this function. In our case, we simply assume a constant-velocity model. Our implementation successively accumulates the relative motions into a common reference frame, resamples these absolute poses using a naïve weighted average approach, then calculates the new relative motions.

B. Means and covariances

The next task is to determine how the mean and covariance are affected when propagated through the nonlinear function f. Fortunately, such a problem has already been addressed successfully by the Unscented Transform (UT). We do not use a UKF, but we employ the UT to estimate how the mean and covariance propagate through a nonlinear function. Whereas the UKF must propagate the state forward in time and predict measurements, we estimate the mean and covariance of our relative measurements after interpolation.

To avoid sampling non-local effects due to a possibly large state space, we use the Scaled UT (SUT) [10]. Like the UT, the SUT estimates the mean and covariance by applying f to $2n+1$ "sigma points" (where n is the dimension of the space). These deterministic sigma points χ_i are centered about the mean of, and discretely approximate, the original distribution:

$$\chi_i = \begin{cases} \overline{v_A} & i = 0 \\ \overline{v_A} + \langle \sigma \rangle_i & 1 \leq i \leq n \\ \overline{v_A} - \langle \sigma \rangle_{i-n} & n+1 \leq i \leq 2n \end{cases} \quad (14)$$

where $\sigma = \sqrt{n\alpha^2 \Sigma_A}$ is calculated with the Cholesky factorization and we set $\alpha = 10^{-2}$ so as to mitigate extreme effects away from the mean [10]. The $\langle \cdot \rangle_i$ operator extracts the i-th column from its argument.

The sigma points are passed through the function, $\Upsilon_i = f(A, \chi_i, B)$, and the new mean and covariance are calculated:

$$\begin{array}{c} \overline{v_B} = \sum_{i=0}^{2n} W_{m,i} \Upsilon_i \\ \Sigma_B = \sum_{i=0}^{2n} W_{c,i} \left(\Upsilon_i - \overline{v_B}\right) \left(\Upsilon_i - \overline{v_B}\right)^T \end{array} \quad (15)$$

As in [10], we use the sigma point weights with $\beta = 2$:

$$\begin{array}{cc} W_{m,0} = 1 - \frac{1}{\alpha^2} & W_{m,i} = \frac{1}{2n\alpha^2} \\ W_{c,0} = W_{m,0} + \left(\beta + 1 - \alpha^2\right) & W_{c,i} = W_{m,i} \end{array} \quad (16)$$

Using this method, we can interpolate the mean and covariance of our incremental pose estimates at new times. It is important to remember, however, that the resulting covariances are only approximations. The interpolation process is nonlinear and, as a result, the transformed distribution (approximated by Υ_i) is not necessarily Gaussian, yet is forced to be so.

VIII. PRACTICAL COVARIANCE MEASUREMENTS

In order to estimate the calibration, our technique requires the covariances of the incremental poses; the inverse covariances serve as observation weights during LSE optimization (Equation 11). Observations with low uncertainty should have heavier weight than those with higher uncertainty. Thus we must accurately characterize sensor noise in order to recover valid calibration parameters.

In some situations, it may be possible to confidently calculate the covariance matrix for each observation. For example, many GPS systems report uncertainty based on time skews and other well-understood physical quantities. On the other hand, estimating the accuracy of the output of a scan-matching algorithm can be difficult. Even with good sensor models, the variance may depend on the environment. For example, an empty room with many right angles may support more accurate localization than a room with lots of clutter.

Ideally, we would like to drive the robot repeatedly along exactly the same path, collect observations at the exact same elapsed times during each run, and compute the variance of the incremental pose observations. However, it is impractical to repeat exactly the same path and timing in an unprepared environment. Instead, we outline a technique that does not require specialized equipment and can be easily employed in the field. The idea is to drive the robot along any path, but

pause periodically. During each pause, we collect many observations from each sensor. By randomly drawing an observation made during each pause, we can effectively generate multiple experiments along a single path.

As an example, consider two planar LIDARs attached to a mobile robot. Fig. 5 shows sample pose estimates from the two sensors. Notice that the density of estimates increases at intervals, where we paused the robot; we refer to the observations within each pause as a "cluster."

Off-line, we approximated the covariance of the incremental poses between clusters. For each pair of adjacent clusters, we selected a random LIDAR scan from each cluster, and used scan-matching to find the relative motion between the two scans. By repeating this drawing and scan matching, we assembled a set of relative motions, the covariance of which approximates the true incremental pose covariance.
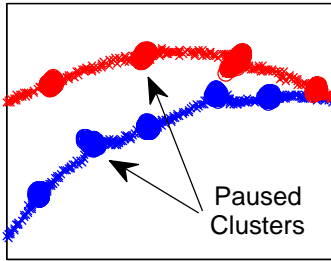

Fig. 5: Observations drawn from each paused interval can be compared to estimate incremental pose covariances off-line.

This technique is appropriate only when inter-cluster incremental poses can be recovered solely from observations drawn from clusters. So, for example, scan-matching from LIDAR data or egomotion estimation from machine vision could provide suitable relative motion measurements, but inertial or odometry data alone would not suffice.

IX. RESULTS

A. Simulation

In order to assess the quality of our calibration estimate and validate the CRLB, we simulated a robot traveling such that sensor r moved along the path shown in Fig. 6. The path had 400 incremental pose observations, and the path for sensor s was offset by various calibrations during the simulations. The 3×3 covariance for each observation was fixed, but varied across observations with a magnitude of 1%-6% of the incremental pose vector (this percentage range was chosen arbitrarily and is not necessarily representative of any particular sensor).

Thirty different calibrations were simulated, with k drawn randomly and uniformly from $[\pm 3\,\mathrm{m}, \pm 3\,\mathrm{m}, \pm 180°]$. In order to validate the CRLB, we executed 200 Monte Carlo simulations, sampling each incremental pose vector from its Gaussian distribution. Fig. 7 shows the results of one such trial with $k = [-0.41\,\mathrm{m}, 1.17\,\mathrm{m}, -162°]$. The mean and standard deviations from the simulations match well with the true value and the CRLB, respectively.
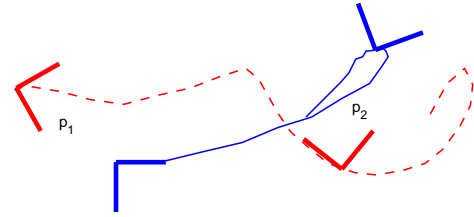

Fig. 6: A simulated travel path for sensor r (blue), with calibration $k = [-0.3\,\mathrm{m}, -0.4\,\mathrm{m}, 30°]$ applied to sensor s (red). Two example sensor frames are shown at p_1 and p_2.
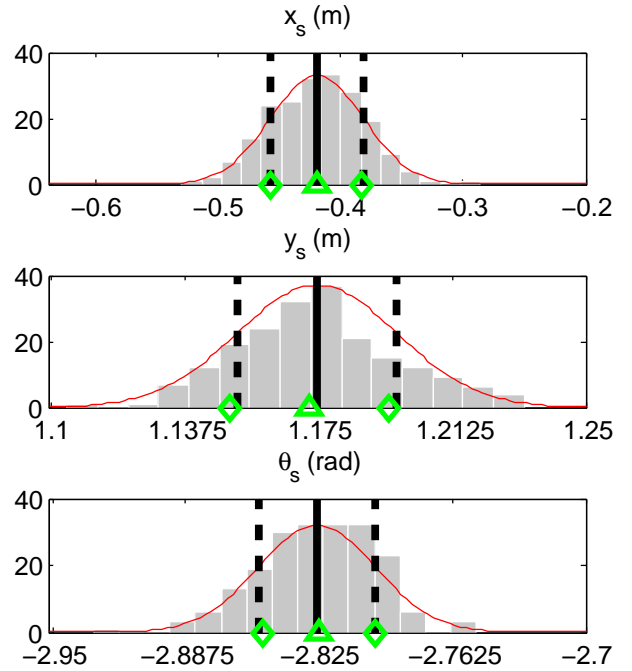

Fig. 7: Histograms (gray) of calibration estimates from 200 simulations of the path in Fig. 6 match well with truth (triangles) and the CRLB (diamonds). Vertical lines indicate mean (solid) and one standard deviation (dashed).

As the number of simulations increases, we expect that the variance of the calibration will be the lowest attainable variance (i.e., the CRLB). With only 200 simulations, we hope that the standard deviation of our calibration estimates will lie near the CRLB. Indeed, Fig. 8 shows that the standard deviation of x_s versus the calculated CRLB for x_s generally agree. Analogous plots for y_s and θ_s (not shown) demonstrate similar agreement.

The scale of the standard deviations (i.e., the scale of the y-axis in Fig. 8 and the x-axes in Fig. 7) are arbitrary since we arbitrarily chose the covariance matrices for the observations. Fig. 9 shows the relationship between the variance of the observations and the CRLB. As expected, as observation noise increases, the CRLB also increases. This indicates that as less information is present in the data, less can be deduced about the calibration. Note that this particular numerical relationship holds only for the path shown in Fig. 6. Other paths will produce different CRLBs (e.g., a degenerate path would produce
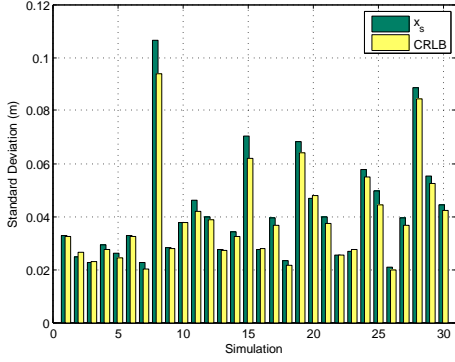
Fig. 8: For 30 different simulated calibration runs, the parameter standard deviation (green) and CRLB (yellow) match well.
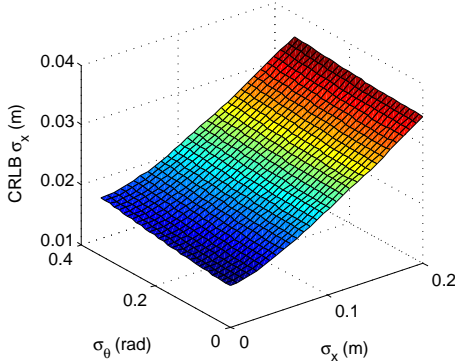

Fig. 9: The CRLB decreases with observation noise.

an "infinite" CRLB).

B. Real data

We attached a $30\,\mathrm{cm} \times 120\,\mathrm{cm}$ machined plate to a small, differentially-steered mobile robot. The plate included a grid of holes, allowing two Hokoyu UTM-30LX LIDARs to be mounted with various "ground truth" calibrations.
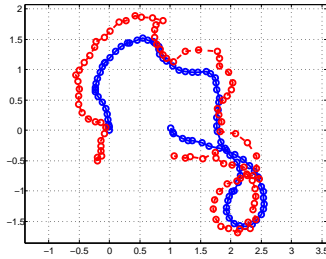

Fig. 10: Paths of r and s when $k = [-0.2\,\mathrm{m}, -0.5\,\mathrm{m}, -120°]$.

We drove the robot manually along a non-degenerate path (Fig. 10), paused every few centimeters, calculated the mean and covariances of 88 incremental pose observations (§ VIII), and estimated the calibration. To validate the CRLB for real data (a step which would not be necessary in general), we sampled the paused clusters to create multiple samplings along the same physical robot path, then estimated incremental poses by scan-matching line features between successive scans. No historical data was maintained (e.g., SLAM was not used).

Fig. 11 shows the distribution of errors from the $k = [-0.2\,\mathrm{m}, -0.5\,\mathrm{m}, -120°]$ calibration; Table I shows results
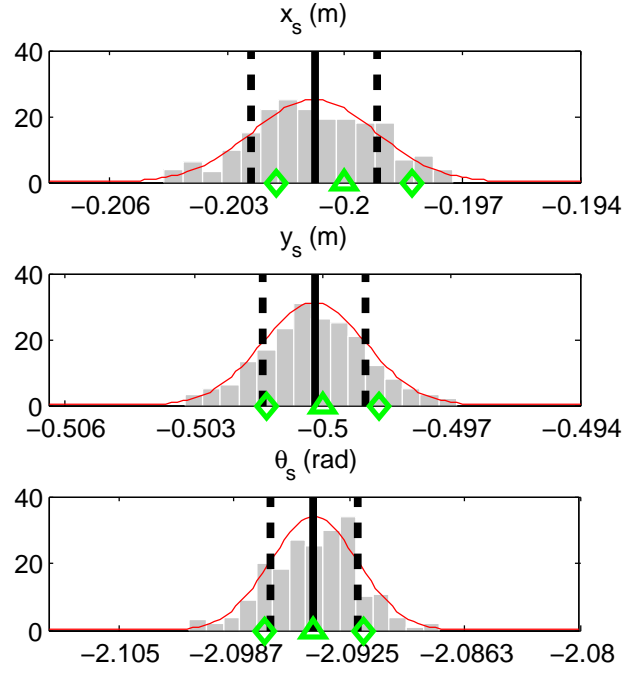

Fig. 11: Estimates from 200 trials using real path of Fig. 10.

from real data for four different sensor calibrations. The average error was at most 1.19 mm (and often much less than 1 mm) and less than approximately 1 milliradian. The standard deviation of the estimates was within 0.2 mm and 0.4 milliradians of the CRLB. Thus, both the calibration estimates and the CRLB agree well with the experimental data.

TABLE I: Calibrations recovered from real data

True Calibration			Estimation Error[a]			CRLB Error[b]		
x (m)	y (m)	θ (deg)	x (mm)	y (mm)	θ (mrad)	x (mm)	y (mm)	θ (mrad)
-0.2	-0.5	0	-0.19	0.03	0.37	-0.14	-0.13	-0.24
-0.2	-0.5	-90	-1.19	0.64	-1.06	-0.03	-0.12	-0.18
-0.2	-0.5	-120	-0.78	-0.20	0.00	-0.16	-0.09	-0.34
-0.2	-0.2	-60	-0.05	0.49	-0.73	-0.05	-0.21	-0.20

[a] Difference between mean of the estimates and the true calibration.
[b] Difference between standard deviation of the estimates and the CRLB.

We also validated the algorithm by finding the calibration between the Hokoyu LIDARs r and s and the robot frame u. We used odometry from wheel encoders, and rotation rate data from an Xsens MTi IMU, to estimate the robot frame's translational and rotational incremental motions respectively. We manually measured the robot frame's calibration parameters to be $k = [-0.08\,\mathrm{m}, -0.06\,\mathrm{m}, 90°]$ at the center of the axle, with a measurement accuracy of about 1 cm and $5°$. We used the procedure in § VIII to estimate observation covariances for the LIDARs, and adopted reasonable values for the covariances of the robot frame incremental poses. Our method estimated the transformation from r to u as $[-0.081\,\mathrm{m}, -0.063\,\mathrm{m}, 90.3°]$ and from r to s to u as $[-0.080\,\mathrm{m}, -0.061\,\mathrm{m}, 90.3°]$, in close agreement with manual measurement.

X. Conclusion

We demonstrate a set of techniques to recover the 3-DOF rigid-body transformation between two coplanar sensors. Our contributions include a novel method to estimate the calibration without relying on accurate sensor localization or any specific preparation of the environment. We showed observability of the system, including an analysis of degenerate conditions, without assuming accurate initial pose estimates or globally consistent sensor observations. The CRLB provides a confidence limit on the estimated parameters, which can be used to design robot paths and incorporated into subsequent algorithms. From a practical perspective, we presented means to interpolate observations and covariances of unsynchronized sensors. These techniques enabled accurate estimation of calibration parameters in both simulated and real experiments.

Our future work will involve recovering the full 6-DOF calibration, given 6-DOF incremental pose observations. The framework presented should generalize to the 6-DOF case, but characterizing the degenerate conditions and optimization in $SO(3) \times \mathbb{R}^3$ will require additional analysis.

Appendix

Algorithm 1 shows pseudocode, with `Optimize` some non-linear least-squares optimizer of the matrix-valued `Cost` function given below, and `JacobianOfG` the Jacobian of G (see Equation 1) evaluated at the estimate $\hat{x}$. Matlab code is posted at `rvsn.csail.mit.edu/calibration`.

Algorithm 1: Calibrate $(z_r, \Sigma_r, t_r, z_s, \Sigma_s, t_s)$

---

inputs : z_r, z_s observed incremental motions of
sensors r, s (resp.) as $3N \times 1$ matrices
Σ_r, Σ_s $3N \times 3N$ covariances of z_r, z_s
t_r, t_s sample times of z_r, z_s

outputs: Calibration estimate $\hat{k}$
Cramer-Rao Lower Bound $CRLB$

$z_s', \Sigma_s' \leftarrow$ `Interpolate`$(t_s, z_s, \Sigma_s, t_r)$
$\Sigma \leftarrow$ `blkdiag`(Σ_r, Σ_s')
$\Psi \leftarrow$ `chol`(Σ^{-1})
$x_0 \leftarrow [v_{r0}, k_0] = [z_r, 0, 0, 0]$ `// initial guess`
$\hat{x} \leftarrow$ `Optimize`$(x_0, z_r, z_s', \Psi, $ `Cost`$)$
$J_G \leftarrow$ `JacobianOfG`$(\hat{x})$
$J \leftarrow J_G^T \Sigma^{-1} J_G$
$\hat{k} \leftarrow \hat{x}_{3N-2:3N}$ `// extract 3-vector and`
$CRLB \leftarrow J_{(3N-2:3N, \, 3N-2:3N)}^{-1}$ `// 3 × 3 matrix`

Function Interpolate$(A, \overline{v_A}, \Sigma_A, B)$

Create χ according to Equation 14
for $\chi_i \in \chi$ **do**
   $\Upsilon_i \leftarrow f(A, \chi_i, B)$ `// see § VII`
Calculate $\overline{v_B}$ and Σ_B according to Equation 15
return $\overline{v_B}, \Sigma_B$

Function Cost(x, z_r, z_s', Ψ)

$v_r, k \leftarrow x$
$\Delta \leftarrow \begin{bmatrix} z_r \\ z_s' \end{bmatrix} - \begin{bmatrix} v_r \\ G(v_r, k) \end{bmatrix}$
return $(\Psi^T \Delta)$ `// of dimension 6N × 1`

References

[1] Y. Bar-Shalom, T. Kirubarajan, and X. Li. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, Inc., New York, NY, USA, 2002.

[2] P. Besl and H. McKay. A method for registration of 3-D shapes. *IEEE PAMI*, 14(2):239–256, February 1992.

[3] J.L. Blanco, F.A. Moreno, and J. Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27:327–351, 2009.

[4] J. Box. Bias in nonlinear estimation. *Journal of the Royal Statistical Society*, 33:171–201, 1971.

[5] A. Censi, L. Marchionni, and G. Oriolo. Simultaneous maximum-likelihood calibration of odometry and sensor parameters. In *Proc. ICRA*, 2009.

[6] S. Ceriani, G. Fontana, A. Giusti, et al. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27:353–371, 2009.

[7] I. Dryden, A. Koloydenko, and D. Zhou. Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics*, 3(3):1102–1123, 2009.

[8] C. Gao and J. Spletzer. On-line calibration of multiple LIDARs on a mobile vehicle platform. In *Proc. ICRA*, pages 279–284, 2010.

[9] E. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *IJRR*, October 2010.

[10] S. Julier. The scaled unscented transformation. In *Proc. ACC*, volume 6, pages 4555–4559, 2002.

[11] J. Kelly and G. Sukhatme. Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration. In *Proc. ICRA*, pages 360–368, 2009.

[12] J. Levinson and S. Thrun. Unsupervised calibration for multi-beam lasers. In *Proc. ISER*, 2010.

[13] A. Martinelli, D. Scaramuzza, and R. Siegwart. Automatic self-calibration of a vision system during robot motion. In *Proc. ICRA*, May 2006.

[14] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.

[15] H. Van Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, New York, 1968.

[16] Z. Wang and G. Dissanayake. Observability analysis of SLAM using Fisher information matrix. In *Proc. IC-CARV*, pages 1242–1247, December 2008.